

JG800/S.5157
09/09/2009



대한민국특허청
KOREAN INDUSTRIAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

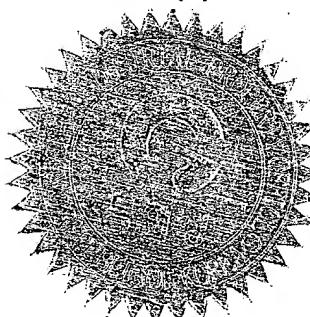
This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

출원번호 : 특허출원 1999년 제 45786 호
Application Number

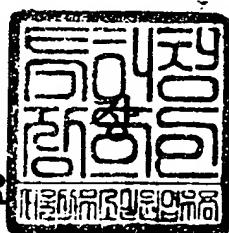
출원년월일 : 1999년 10월 21일
Date of Application

출원인 : 삼성전자 주식회사
Applicant(s)

2000년 02월 15일



특허청
COMMISSIONER



CERTIFIED COPY OF

PRIO/PRIORITY DOCUMENT

1019990045786

2000/2/1

【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【참조번호】	0002		
【제출일자】	1999. 10. 21		
【발명의 명칭】	분기 예측 정확도 히스토리를 이용한 분기 예측기		
【발명의 영문명칭】	BRANCH PREDICTOR USING BRANCH PREDICTION ACCURACY HISTORY		
【출원인】			
【명칭】	삼성전자 주식회사		
【출원인코드】	1-1998-104271-3		
【대리인】			
【성명】	임창현		
【대리인코드】	9-1998-000386-5		
【포괄위임등록번호】	1999-007368-2		
【대리인】			
【성명】	권혁수		
【대리인코드】	9-1999-000370-4		
【포괄위임등록번호】	1999-056971-6		
【발명자】			
【성명의 국문표기】	조장호		
【성명의 영문표기】	CHO, JANG HO		
【주민등록번호】	690416-1402723		
【우편번호】	441-340		
【주소】	경기도 수원시 권선구 구운동 462 삼환APT 3동 111호		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 임창현 (인) 대리인 권혁수 (인)		
【수수료】			
【기본출원료】	17	면	29,000 원
【가산출원료】	0	면	0 원

1019990045786

2000/2/1

【우선권주장료】 0 건 0 원

【심사청구료】 8 황 365,000 원

【합계】 394,000 원

【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

여기에 개시된 분기 예측기는, 분기 예측 수단과, 분기 예측 수단으로부터 예측된 예측 조건 검색 결과의 정확도를 저장하기 위한 정확도 히스토리 테이블을 포함한다. 본 발명에 의한 분기 예측기는, 상기 분기 예측 수단으로부터의 예측 조건 검색 결과를 직접 사용하지 않고, 분기 예측 정확도 히스토리에 기반을 둔 예측 조건 검색 결과를 최종 분기 예측 값으로 사용한다. 그 결과, 2-레벨 분기 예측시 발생하는 분기 예측 실패률 줄일 수 있고, 더욱 효과적인 CPU의 명령 실행을 제공할 수 있다.

【대표도】

도 2

【명세서】

【발명의 명칭】

분기 예측 정확도 히스토리를 이용한 분기 예측기{BRANCH PREDICTOR USING BRANCH PREDICTION ACCURACY HISTORY}

【도면의 간단한 설명】

도 1은 종래의 2-레벨 분기 예측기의 구조를 보여주기 위한 도면; 그리고
도 2는 본 발명의 바람직한 실시예에 의한 2-레벨 분기 예측기의 구조를 보여주기
위한 도면.

도면의 주요 부분에 대한 부호의 설명

10, 15 : 브랜치 히스토리 레지스터 20, 25 : 패턴 히스토리 테이블

30 : 제 1 논리 회로 40 : 비교기

50 : 제 2 논리 회로 60 : 정확도 히스토리 테이블

70 : 멀티플렉서

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<8> 본 발명은 CPU의 분기 예측기에 관한 것으로, 좀 더 구체적으로는 2-레벨 분기 예측시 발생되는 분기 예측 실패를 줄이기 위해 분기 예측 정확도 히스토리를 이용하는 분기 예측기에 관한 것이다.

<9> 조건 분기문(conditional branch)이란, 조건 검사 과정을 거친 후 조건이 참이면

분기문에서 명시해 주는 목적 주소(target address)의 명령어를 수행하고, 조건이 거짓일 경우는 분기문 다음의 명령어를 수행하는 프로그램 명령어의 일종이다. 예를 들어, '만약 A > B 이면 명령어 X로 분기하라'라는 조건 분기문이 있을 경우, 만약 A가 B보다 크다면 프로그램 제어 수순은 명령어 X로 분기하게 될 것이고, 만약 A가 B보다 크지 않다면 프로그램 제어 수순은 분기하지 않고 분기문 다음에 있는 명령어를 수행할 것이다. 이러한 조건 분기문은, 조건 분기문 다음의 명령어를 결정하기 위해 조건을 검사하는 과정이 필요하다. 이는, 파이프라인(pipeline) 방식을 채택함에 의해 빠른 명령어의 페치(fetch)를 필요로 하는 근래의 CPU(Central Processing Unit)에서 성능을 저하시키는 원인이 된다.

- <10> 최근의 CPU는 이러한 문제를 해결하기 위해서, 조건 분기문을 만날 경우 조건에 대한 결과를 미리 예측하기 위한 분기 예측기(branch predictor)를 사용한다. 분기 예측기는, 분기 명령어를 만날 경우 조건 검사 과정의 결과가 나오기 전에 이전의 분기 명령어의 결과를 사용하여 해당 분기 명령어의 조건 검사 결과를 예측한다. CPU는 예측된 조건 검사 결과에 따라서 다음의 명령어를 페치하여 수행한다. 그 결과, 파이프라인의 정지(stall)를 없앨 수 있다. 그러나, 만약 분기 예측의 결과가 틀릴 경우(branch prediction miss)에는, 이미 페치하여 수행 중인 명령어들의 진행을 중지시키고 실제적인 다음 명령어를 페치하여 수행해야 하는 오류로 인한 불이익(miss penalty)이 발생한다.
- <11> 이러한 분기 명령의 실패율(miss ratio)을 줄이기 위해 다양한 종류의 분기 예측기들이 제안되어 오고 있는데, 그 중에서 가장 광범위하게 사용되는 분기 예측

기는 2-레벨 분기 예측기(2-level branch predictor)이다. Intel사의 P6 프로세서는 정확도를 개선하기 위한 2-레벨 분기 예측 알고리즘을 최초로 사용하였다. Tse-Yu Yeh와 Yale Patt에 의해 최초로 발표된 이 2-레벨 분기 예측 알고리즘은 오늘날 최적의 프로세서에 의해서 90% 이상의 정확도를 끌어낼 수 있는 잠재력을 가진다.

<12> 도 1은 종래의 2-레벨 분기 예측기의 구조를 보여주기 위한 도면이다. 이 도면은, Linley Gwennap에 의해 1995년 3월 27일, MICROPROCESSOR REPORT의 17-21 페이지에 발표된 논문 'New Algorithm Improves Branch Prediction'의 도 2를 예로 든 것이다. 도 1을 참조하면, 상기 2-레벨 분기 예측기는 크게 브랜치 히스토리 레지스터(Branch History Register ; BHR)(10)와 패턴 히스토리 테이블(Pattern History Table ; PHT)(20)로 구성된다.

<13> 상기 브랜치 히스토리 레지스터(10)는 k 개의 이전 분기 명령어의 조건 검색 결과를 저장하는 레지스터로서, 상기 레지스터에 저장된 '1'은 조건 검색 결과가 '참'인 경우를, 그리고 상기 레지스터에 저장된 '0'은 조건 검색 결과가 '거짓'인 경우를 각각 나타낸다. 이와 같이, 이전에 수행된 k 번의 분기 명령어의 조건 검색 결과를 패턴(pattern)이라 한다.

<14> 상기 패턴 히스토리 테이블(20)은, 상기와 같은 각각의 패턴에 따라서 다음에 수행될 분기 명령어의 조건 검사 결과를 미리 예측하기 위한 정보인 패턴 히스토리 비트(Sc)를 저장하는데 사용된다. 예를 들어, 상기 2-레벨 분기 예측기는 임의의 패턴 '111010'에 대응되는 패턴 히스토리 비트(Sc) '10'에 의해 예측 조건 검색 결과(I(Sc))를 발생하며, 이 예측 조건 검색 결과(I(Sc))에 의해 분기 명령어

다음의 명령어 폐치가 이루어지게 된다. Linley Gwennap에 의한 상기 논문을 참조하면, 이 2-레벨 분기 예측기의 예측 조건 검색 결과(I(Sc))는 패턴 히스토리 비트(Sc)의 최상위 비트(most significant bit ; MSB)에 의해 결정된다.

- <15> 예를 들어, 분기 명령의 실제 조건 검사 결과를 Rc라 할 때, 이 값이 예측 조건 검색 결과(I(Sc))와 다를 경우 이를 분기 예측 실패(prediction miss)라 한다. 이 경우, 상기 예측 조건 검색 결과(I(Sc))에 의해 분기 명령어 이후에 폐치 되어 수행 중인 명령어들을 모두 취소시켜야 한다.
- <16> 이와 같은 분기 명령의 실제 조건 검사 결과(Rc)에 의해 상기 브랜치 히스토리 레지스터(10)의 값과 상기 패턴 히스토리 테이블(20)에 저장되어 있는 패턴 히스토리 비트(Sc)가 변화되는데, 그 과정은 다음과 같다.
- <17> 먼저, 상기 분기 명령의 실제 조건 검사 결과(Rc)가 브랜치 히스토리 레지스터(10)에 저장되면 나머지 비트들은 쉬프트(shift)하게 된다. 이에 따라 해당 패턴의 저장 장소인 패턴 히스토리 테이블(20)에 저장된 패턴 히스토리 비트(Sc)도 상기 실제 조건 검사 결과(Rc) 값에 따라 새로운 정보로 바뀌게 된다. 예를 들어, 실제 조건 검사 결과(Rc)가 '참'을 나타내는 '1'이면 상기 패턴 히스토리 비트(Sc)의 값은 '1' 만큼 증가되고, 실제 조건 검사 결과(Rc)가 '거짓'을 나타내는 '0'이면 상기 패턴 히스토리 비트(Sc)의 값은 '1' 만큼 감소된다.
- <18> 1981년 5월, J. Smith에 의해 8th International Symposium on Computer Architecture, 135-148쪽에 발표된 논문 'A Study of Branch Prediction Strategies'에 설명된 바와 같이, 상기 방법에 있어서 패턴 히스토리 테이블(20)에

저장되어 있는 패턴 히스토리 비트(Sc)는 업/다운 세츄레이팅 카운터(up/down saturating counter)에 의해 구성될 수 있다. 이 세츄레이팅 카운터는 상기 패턴 히스토리 비트(Sc) 값이 최소 값일 때 실제 조건 검사 결과(Rc)가 '거짓'(즉, '0')인 경우가 발생하면 상기 패턴 히스토리 비트(Sc) 값을 최소 값 그대로 유지시켜주고, 그리고 상기 패턴 히스토리 비트(Sc) 값이 최대 값일 때 실제 조건 검사 결과(Rc)가 '참'(즉, '1')인 경우가 발생하면 상기 패턴 히스토리 비트(Sc) 값을 최대 값 그대로 유지시켜 주는 특성을 가진다.

<19> 분기 예측에 대한 정확도가 더 개선될 수 있고, 다른 분기 예측 알고리즘들을 사용함에 의해서 조율될 수 있음에도 불구하고, 분기 예측 실패는 여전히 발생되고 있다. 이 분기 예측 실패가 확인될 때까지, 온당하지 않은 코드 섹션으로부터 많은 명령어들이 명령 실행 파이프라인 내에서 다양한 단계로 처리되는 문제가 발생한다.

<20> 따라서, 이 문제를 해결하기 위한 방안으로는, Sharangpani et al.에 의해 1999년 1월 12일 취득된 U.S. Pat. No. 5,860,017, 'PROCESSOR AND METHOD FOR SPECULATIVELY EXECUTING INSTRUCTIONS FROM MULTIPLE INSTRUCTION STREAMS INDICATED BY A BRANCH INSTRUCTION' 등이 있다. Sharangpani et al.은, 조건 검사 결과가 '참'인 경우와 '거짓'인 경우 모두에 대한 명령을 실행한다. 그리고, 처음 수행되는 분기 명령어, 최근 예측이 잘못된 경우, Y번 수행 중 X번 예측이 잘못된 경우, 그리고 잘못된 예측 후 일정 시간 지나서 이를 다시 수행하는 경우를 예측된 정확도가 희박한 경우로 판단한다. 그러나, 이 방법에 있어서 리소스(resource) 충돌(conflict)로 인한 성능 저하의 가능성이 있을 뿐만 아니라, 하드웨어 코스트(hardware cost)가 높은 단점이 있다.

<21> 따라서, 상기와 같은 분기 예측 실패를 줄임으로써 보다 효과적인 명령 실행을 제

공할 수 있는, 간단한 회로적 구성과 낮은 하드웨어 코스트를 가지는 분기 예측기가 요구된다.

【발명이 이루고자 하는 기술적 과제】

<22> 따라서, 본 발명의 목적은 상술한 제반 문제점을 해결하기 위해 제안된 것으로, CPU의 2-레벨 분기 예측기에 있어서, 간단한 회로적 구성과 낮은 하드웨어 코스트로 분기 예측 실패를 줄일 수 있는 장치를 제공하는데 있다.

【발명의 구성 및 작용】

<23> 상술한 바와 같은 본 발명의 목적을 달성하기 위한 본 발명의 특징에 의하면, 분기 예측기는, 분기 명령어의 조건 검사 결과를 예측하기 위한 분기 예측 수단, 상기 분기 예측 수단으로부터 예측된 상기 분기 명령어의 예측 조건 검색 결과와 상기 분기 명령어의 실제 조건 검색 결과를 비교한 비교 신호를 발생하기 위한 비교기, 상기 예측 조건 검색 결과의 분기 예측 정확도 히스토리를 저장하기 위한 정확도 히스토리 테이블, 상기 비교 신호에 응답해서 상기 정확도 히스토리 테이블에 저장될 정확도 히스토리 비트를 발생하기 위한 제 1 논리 회로, 그리고 상기 정확도 히스토리 비트에 의해 발생되는 정확도 예측 신호에 응답해서 상기 분기 예측 수단으로부터 발생되는 상기 예측 조건 검색 결과와 반전된 상기 예측 조건 검색 결과 중 어느 하나를 분기 예측 값으로 출력하기 위한 멀티플렉서를 포함한다.

<24> 상기 분기 예측 수단은, 이전 분기 명령어들의 조건 검사 결과들을 저장하기 위한 브랜치 히스토리 레지스터, 상기 조건 검사 결과 패턴에 대응되는 예측 조건 검색 결과를 발생하기 위한 패턴 히스토리 비트를 저장하기 위한 패턴 히스토리 테이블, 그리고

상기 실제 조건 검사 결과에 응답해서 상기 패턴 히스토리 비트를 발생하기 위한 제 2 논리 회로를 포함한다.

<25> (실시 예)

<26> 이하 본 발명에 따른 실시예를 첨부된 도면을 참조하여 상세히 설명한다.

<27> 본 발명의 신규한 분기 예측기는, 분기 예측 수단에 의해 얻어진 예측 조건 검색 결과를 직접 사용하지 않고, 분기 예측 정확도 히스토리에 기반을 둔 예측 조건 검색 결과를 최종 분기 예측 값으로 사용한다. 그 결과, 2-레벨 분기 예측시 발생하는 분기 예측 실패를 줄일 수 있고, 더욱 효과적인 CPU의 명령 실행을 제공할 수 있다.

<28> 도 2는 본 발명의 바람직한 실시예에 의한 2-레벨 분기 예측기의 구조를 보여주기 위한 도면이다. 도 2를 참조하면, 본 발명에 의한 2-레벨 분기 예측기는, 이전 분기 명령어들의 조건 검사 결과들을 저장하기 위한 브랜치 히스토리 레지스터(15), 예측 조건 검색 결과(I(Sc))를 발생하기 위한 패턴 히스토리 비트(Sc)를 저장하기 위한 패턴 히스토리 테이블(25), 그리고 상기 예측 조건 검색 결과(I(Sc))의 분기 예측 정확도 히스토리를 저장하기 위한 정확도 히스토리 테이블(60)을 포함한다. 상기 정확도 히스토리 테이블(60)은 메모리 어레이(memory array)로 구성된다.

<29> 상기 패턴 히스토리 테이블(25)에는 실제 조건 검사 결과(Rc)에 응답해서 상기 패턴 히스토리 테이블(25)에 저장될 패턴 히스토리 비트(Sc)를 발생하기 위한 제 1 논리 회로(30)가 연결되고, 상기 정확도 히스토리 테이블(60)에는 상기 정확도 히스토리 테이블(60)에 저장될 정확도 히스토리 비트(Ac)를 발생하기 위한 제 2 논리 회로(50)가 연결된다.

<30> 본 발명에 의한 분기 예측기는, 상기 패턴 히스토리 비트(Sc)에 의해 발생된 예측 조건 검색 결과(I(Sc))와 상기 분기 명령어의 실제 조건 검색 결과가 동일한지 여부를 비교하여 그 결과로서 비교 신호를 발생하기 위한 비교기(40)를 포함하는데, 이 비교 신호는 상기 제 2 논리 회로(50)에 입력되어 정확도 히스토리 비트(Ac)를 발생하는데 사용된다. 그리고 본 발명에 의한 분기 예측기는, 최종 분기 예측 값으로서 상기 예측 조건 검색 결과(I(Sc))와 반전된 예측 조건 검색 결과 중 어느 하나를 선택하여 출력하기 위한 멀티플렉서(70)를 포함한다. 이 멀티플렉서(70)의 선택 신호로는, 상기 정확도 히스토리 비트(Ac)에 의해 발생된 정확도 예측 신호(I(Ac))가 사용된다.

<31> 본 발명에 의한 분기 예측기의 동작을 살펴보면 다음과 같다.

<32> 먼저, 브랜치 히스토리 레지스터(15)에 저장된 패턴에 대응되는 패턴 히스토리 비트(Sc)에 의해서 예측 조건 검색 결과(I(Sc))가 발생된다. 이 예측 조건 검색 결과(I(Sc))는 비교기(40)에 입력되어 실제 조건 검색 결과(Rc)와 비교된다.

<33> 상기 실제 조건 검색 결과(Rc)는 분기 명령어의 조건에 대한 '참' 또는 '거짓'의 여부에 따라 '1' 또는 '0'의 값을 가지며, 이 값에 따라 브랜치 히스토리 레지스터(15)의 값이 갱신된다. 그리고, 이 갱신된 브랜치 히스토리 레지스터(15)의 값에 따라 패턴 히스토리 비트(Sc)의 값 또한 갱신되는데, 제 1 논리 회로(30)가 바로 이 패턴 히스토리 비트(Sc)의 값을 갱신하는 역할을 수행한다. 상기 제 1 논리 회로(30)는 상기 실제 조건 검색 결과(Rc)가 '참'(즉, '1')일 때 패턴 히스토리 비트(Sc)의 값을 '1' 만큼 증가하고, 상기 실제 조건 검색 결과(Rc)가 '거짓'(즉, '0')일 때 패턴 히스토리 비트(Sc)의 값을 '1' 만큼 감소해주는 업/다운 세츄레이팅 카운터로 구성된다.

<34> 상기 예측 조건 검색 결과(I(Sc))는 패턴 히스토리 테이블(25)에 저장되어 있는 패

턴 히스토리 비트(Sc)의 최상위 비트(MSB)에 의해서 '1' 또는 '0'의 분기 예측 값을 가진다. 상기 비교기(40)는 이 실제 조건 검색 결과(Rc)와 예측 조건 검색 결과(I(Sc))가 서로 일치하는지 여부를 판별해서 두 검색 결과(Rc, I(Sc))가 서로 일치할 경우에는 '1'의 값을, 그리고 두 검색 결과(Rc, I(Sc))가 서로 일치하지 않을 경우에는 '0'의 값을 비교 신호로서 출력한다.

<35> 이와 같은 방법에 의해서 발생된 '1' 또는 '0'의 값을 가지는 비교 신호를 받아들인 제 2 논리 회로(50)는, 이 비교 신호를 바탕으로 하여 정확도 히스토리 테이블(60)에 저장될 정확도 히스토리 비트(Ac)의 값을 결정한다. 이 제 2 논리 회로(50)는 업/다운 세 츠레이팅 카운터로 구성되어, 만약 상기 비교 신호가 '1'이면(즉, 두 검색 결과(Rc, I(Sc))가 서로 일치할 경우에는) 정확도 히스토리 비트(Ac)의 값을 1 증가하여 저장하고, 만약 상기 비교 신호가 '0'이면(즉, 두 검색 결과(Rc, I(Sc))가 서로 일치하지 않을 경우에는) 정확도 히스토리 비트(Ac)의 값을 1 감소하여 저장한다. 이 정확도 히스토리 비트(Ac)의 값은, 일정 시간 모니터링(monitoring)한 후, 해당 패턴의 분기 예측의 정확도를 학습시킨 후 사용할 수도 있다.

<36> 앞에서 설명한 바와 같은 일련의 과정에 의해서 상기 정확도 히스토리 테이블(60)에 저장될 정확도 히스토리 비트(Ac)의 값이 정해지게 되고, 바로 이 정확도 히스토리 비트(Ac)에 의해서 본 발명에 의한 분기 예측기에서 발생되는 예측 값이 정확한지 여부를 예측하게 된다.

<37> 예를 들어, 브랜치 히스토리 레지스터(15)에 저장된 패턴 '11...10'에 대응되는 패턴 히스토리 비트(Sc)가 '011'일 때, 이에 대응되는 정확도 히스토리 비트(Ac)의 최상위 비트(MSB)에 의해서 정확도 예측 신호(I(Ac))가 발생된다. 이 정확도 예측 신호(I(Ac))

는 분기 예측의 결과가 옳은지 여부를 예측하기 위한 신호이다. 예를 들어, 분기 예측이 정확하다고 예측될 때 상기 정확도 예측 신호(I(Ac))는 '1'의 값을, 그리고 분기 예측이 정확하지 않다고 예측될 때 상기 정확도 예측 신호(I(Ac))는 '0'의 값을 나타낸다. 상기 정확도 예측 신호(I(Ac))는, 최종 분기 예측 값으로서 상기 예측 조건 검색 결과(I(Sc))와 반전된 예측 조건 검색 결과 중 어느 하나를 출력하기 위한 멀티플렉서(70)의 선택 신호로 사용된다.

<38> 따라서, 본 발명에 의한 분기 예측기는, 패턴 히스토리 비트(Sc)에 의해서 예측된 예측 조건 검색 결과(I(Sc))에 의한 브랜치 동작을 바로 수행하지 않고, 예측 조건 검색 결과(I(Sc))와 실제 조건 검색 결과(Rc)의 비교에 의해 산출된 분기 예측 정확도 히스토리를 가지고 있는 정확도 히스토리 비트(Ac)에 의한 정확도 예측 신호(I(Ac))에 의해서 분기 예측 값을 선택하여 사용한다. 그 결과, 분기 예측 정확도 히스토리에 기반을 둔 예측 조건 검색 결과의 선택에 의해서, 2-레벨 분기 예측시 발생하는 분기 예측 실패를 줄일 수 있고, 더욱 효과적인 CPU의 명령 실행을 제공할 수 있다. 뿐만 아니라, 본 발명에 의한 분기 예측기는, 종래의 분기 예측기에 분기 예측이 정확한가를 예측하기 위한 정확도 히스토리 테이블(60)과 멀티플렉서(70) 회로를 추가한 것으로서, 간단한 회로적 구성과 낮은 하드웨어 코스트로 분기 예측 실패를 줄일 수 있다.

<39> 이상에서, 본 발명에 따른 회로의 구성 및 동작을 상기한 설명 및 도면에 따라 도시하였지만 이는 예를 들어 설명한 것에 불과하며 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 다양한 변화 및 변경이 가능함은 물론이다.

【발명의 효과】

<40> 이상과 같은 본 발명에 의하면, 간단한 회로적 구성과 낮은 하드웨어 코스트로 분

1019990045786

2000/2/1

기 예측 실패를 줄일 수 있다.

【특허청구범위】**【청구항 1】**

분기 예측기에 있어서:

분기 명령어의 조건 검사 결과를 예측하기 위한 분기 예측 수단과;

상기 분기 예측 수단으로부터 예측된 상기 분기 명령어의 예측 조건 검색 결과와

상기 분기 명령어의 실제 조건 검색 결과를 비교한 비교 신호를 발생하기 위한 비교기와

;

상기 예측 조건 검색 결과의 분기 예측 정확도 히스토리를 저장하기 위한 정확도 히스토리 테이블과;

상기 비교 신호에 응답해서 상기 정확도 히스토리 테이블에 저장될 정확도 히스토리 비트를 발생하기 위한 제 1 논리 회로; 그리고

상기 정확도 히스토리 비트에 의해 발생되는 정확도 예측 신호에 응답해서 상기 분기 예측 수단으로부터 발생되는 상기 예측 조건 검색 결과와 반전된 상기 예측 조건 검색 결과 중 어느 하나를 최종 분기 예측 값으로 출력하기 위한 멀티플렉서를 포함하는 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【청구항 2】

제 1 항에 있어서,

상기 분기 예측 수단은,

이전 분기 명령어들의 조건 검사 결과들을 저장하기 위한 브랜치 히스토리 레지스터와;

상기 조건 검사 결과 패턴에 대응되는 상기 예측 조건 검색 결과를 발생하기 위한 패턴 히스토리 비트를 저장하기 위한 패턴 히스토리 테이블; 그리고 상기 실제 조건 검사 결과에 응답해서 상기 패턴 히스토리 비트를 발생하기 위한 제 2 논리 회로를 포함하는 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【청구항 3】

제 1 항에 있어서,

상기 정확도 히스토리 테이블은, 메모리 어레이로 구성되는 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【청구항 4】

제 1 항에 있어서,

상기 비교기는, 상기 예측 조건 검색 결과와 상기 실제 조건 검색 결과가 일치하면 논리 '1'의 비교 신호를, 그리고 일치하지 않으면 논리 '0'의 비교 신호를 발생하는 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【청구항 5】

제 1 항에 있어서,

상기 제 1 논리 회로는, 업/다운 세츄레이팅 카운터인 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【청구항 6】

제 5 항에 있어서,

상기 제 1 논리 회로는, 해당 패턴의 분기 예측의 정확도를 학습시킨 후 사용할 수 있는 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【청구항 7】

제 1 항에 있어서,

상기 정확도 예측 신호는, 상기 정확도 히스토리 비트의 최상위 비트에 의해 결정되는 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【청구항 8】

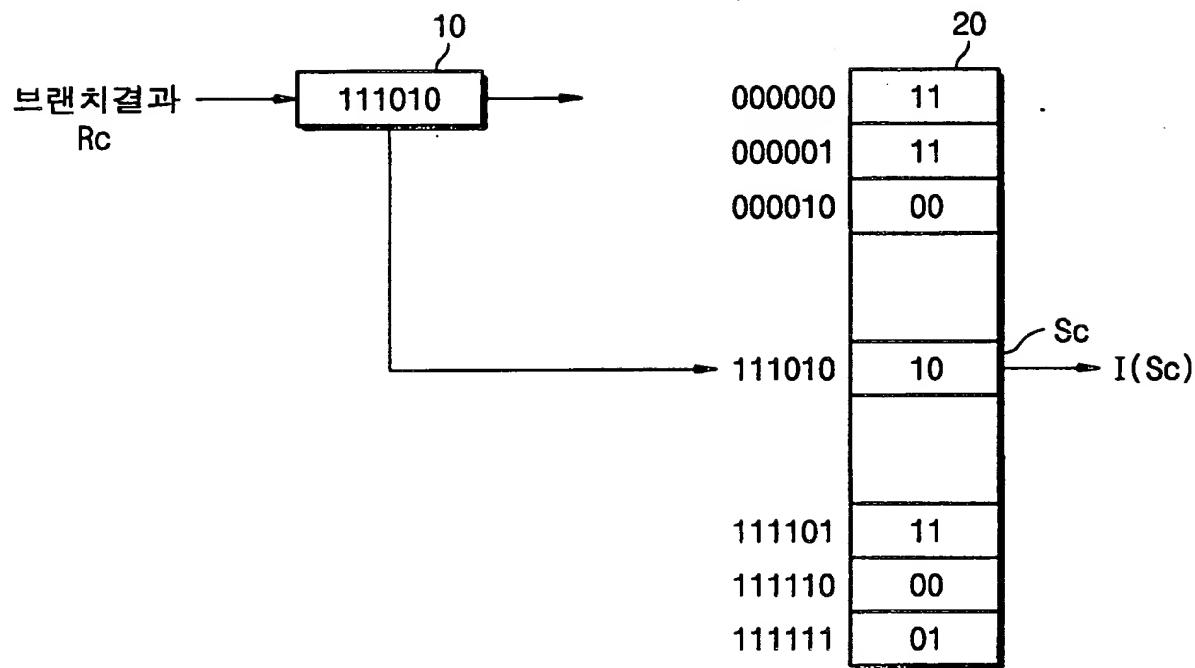
제 2 항에 있어서,

상기 제 2 논리 회로는, 업/다운 세츄레이팅 카운터인 것을 특징으로 하는 분기 예측 정확도 히스토리를 이용한 분기 예측기.

【도면】

【도 1】

(종래기술)



【도 2】

